

Track: CMS Blue Button



INTEROPERABILITY
INSTITUTE

Table of Contents

Introduction to CMS Blue Button.....	3
Scenario 1: Showcasing EOB's and Coverage Info from Silvercare.....	3
Queries for Scenario 1.....	4
Query 1- Finding a patient by SSN.....	4
Query 2- Locating EOB Records for the Patient.....	6
Query 3- Finding Coverage Information for the Patient.....	7
Scenario 2: Navigating Co-Insurance.....	9
Queries for Scenario 2.....	9
Query 1- Finding a patient by SSN (Silvercare).....	9
Scenario 3: Comparing Medication Prices.....	12
Queries for Scenario 3.....	12
Query 1- – Finding the Patient's ID on an EHR PIT	12
Query 2- Finding Medication Requests for a Patient.....	13
Query 3- Finding a Patient by SSN.....	14
Query 4- Finding a Patient's Prescriptions in Their EOB Records.....	15
Appendix A: Steps for Authorization.....	18
Appendix B: Bulk Querying.....	27
Appendix C: Sample Users.....	28
Judging Criteria.....	29

Introduction to CMS Blue Button

Blue Button allows Medicare beneficiaries to quickly and easily access their medical claims data online. It also allows for the safe and secure exchange of beneficiaries' medical claims between the various applications and programs that they interact with and rely on to help them manage their plan.

Blue Button uses HL7® FHIR® and OAuth 2.0 standards for beneficiary data and identification respectively to ensure that access to and exchange of information is easy and secure. Over 53 Million beneficiaries have at least 1 claim in the Blue Button dataset, including historical claims going back 4 years. In addition, Blue Button offers a synthetic data set that's provides developers with the chance to test their applications with accurate data without the risk that comes with using the sensitive Private Health Individuals of real beneficiaries.

It contains parts A (inpatient coverage), B (outpatient coverage), and D (prescription drug coverage) data for beneficiaries with traditional Medicare coverage. It also contains Part D data for those on Medicare Advantage plans.

Important Note: For this track each team will be assigned a patient existing in the IOL platform with a unique SSN and collection of information (located in Appendix C of this document). The examples used in this document will not produce the same information as the persona assigned to your team.

Scenario 1- Showcasing Explanation of Benefits (EOBs) and Coverage info from one PIT (SILVERCARE)

Terry needs to see information on his current Medicare coverage. He is unable to find his Medicare card. He would also like to see a list of his EOBs and access information on how to read them.

Action: Explore ways to showcase a patient's medical insurance information, review coverage, and allow him to review their EOBs.

Precondition: Be able to utilize the queries listed below in the Silvercare PIT in IOL.

Success Criteria: Successfully locate a patient's EOB's and coverage information.

Examples- Possible Queries:

1. Find a patient ID using SSN
2. Locate the patient's EOB's using the *ExplanationOfBenefits* resource.

3. Review the patient's coverage information in the *Coverage* resource

Queries for Scenario 1

Query 1 – Finding A Patient By SSN

Because the HAPI FHIR® Interface only supports a limited set of search parameters, an additional initial query is necessary to find the Patient record by SSN. The id of the Patient record we find will be necessary to execute future queries.

The screenshot shows the HAPI FHIR interface. On the left, a navigation menu lists resources: ExplanationOfBenefit (187100), Coverage (21424), Patient (5932), Account, ActivityDefinition, and AdverseEvent. The 'Patient' resource is highlighted with a red box. On the right, the 'Search' tab is active. A search bar contains a magnifying glass icon and the word 'Search'. Below it, the 'Search Parameters' section has a dropdown menu set to 'Identifier - A patient identifier'. To the right of the dropdown is a 'Matches' dropdown, a 'System' dropdown, and a 'Code' field containing '000003512'. The 'Includes' section below states 'Also include resources which are referenced by the search results'.

Searching for a patient requires selecting the Patient option from the resource's navigation menu on the left side of the page. Specifying the SSN to search for is then accomplished by selecting the identifier option in the search parameters dropdown, then entering the target SSN in the code field to the right (as seen below SSN 000003512 is used in this example).

Note: In order to perform a bulk query of patient data, simply press "Search" without including the patient identifier in the search parameters.

This query produces the result shown below. Pay particular attention to the lower-right area of the screen, which contains the section labelled Result Body. This is the data returned in response to the query.

➤ Request	GET https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient?identifier=000003512&_pretty=true				
Request Headers	Accept-Charset: utf-8 Authorization: Basic aW50ZXJvcF9waXQ6UUNxdWhyblkzcDFSSDRlZTlsejE3cmlyZ2VjeWo0TW9BTf0 Accept: application/fhir+xml;q=1.0, application/fhir+json;q=1.0, application/xml+fhir;q=0.9, application/json+fhir;q=0.9 User-Agent: HAPI-FHIR/4.2.0 (FHIR Client; FHIR 3.0.2/DSTU3; apache) Accept-Encoding: gzip				
⬅ Response	✓ HTTP 200				
Response Headers	x-request-id: wGIDY0QLrvzKWHfu date: Fri, 15 May 2020 15:22:35 GMT last-modified: Fri, 15 May 2020 15:22:35 GMT transfer-encoding: chunked x-powered-by: HAPI FHIR 4.2.0 REST Server (FHIR Server; FHIR 3.0.2/DSTU3) connection: keep-alive content-type: application/fhir+json;charset=UTF-8				
Result Body	<div> <div>Bundle contains 1 / 1 entries</div> <div> <div>JSON bundle</div> <div>(4455 bytes)</div> </div> <table> <thead> <tr> <th>ID</th><th>Updated</th></tr> </thead> <tbody> <tr> <td> <div>Read</div> <div>Update</div> Patient/155/_history/1 </td><td>2020-04-17 20:52:25</td></tr> </tbody> </table> </div>	ID	Updated	<div>Read</div> <div>Update</div> Patient/155/_history/1	2020-04-17 20:52:25
ID	Updated				
<div>Read</div> <div>Update</div> Patient/155/_history/1	2020-04-17 20:52:25				

The Result Body section in this case contains a line at the top that reads “Bundle contains 1 / 1 entries”. This line shows how many results matched the query that was specified; the first number is the count of results included in this response, the second is the total number of record satisfying that query in this PIT. When a query matches a large number of records, the numbers can be different because the FHIR® server has a limit on the number of records it can return in a single response.

Below the summary section is a section labelled Raw Message. This section contains the actual data sought by the query (in this case, a Patient record identified by the supplied SSN).

Raw Message

```
{
  "resourceType": "Bundle",
  "id": "5726bf18-5e32-4979-9b2e-3a8a6dd9c699",
  "meta": {
    "lastUpdated": "2020-05-15T15:22:35.307+00:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient?_pretty=true&identifier=000003512"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient/155",
      "resource": {
        "resourceType": "Patient",
        "id": "155",
        "meta": {
          "extension": [
            {
              "url": "http://hl7.org/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#EbYsLmZVFpgD0k2Q"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:52:25.000+00:00"
        },
        "text": {
          "status": "generated",
          "div": "<div xmlns='http://www.w3.org/1999/xhtml'><div class='hapiHeaderText'>Terry Bryan <b>BRUCE </b></div><table class='hapiPropertyTable'><tbody><tr><td>Identifier</td><td>000003512</td></tr><tr><td>Date of birth</td><td><span>01 September 1948</span></td></tr></tbody></table></div>"
        },
        "extension": [
          {
            "url": "http://mihin.org/extension/copyright",
            "valueString": "Copyright 2014-2020 Michigan Health Information Network Shared Services. Licensed under the Apache License, Version 2.0 (the 'License'); you may not use this file except in compliance with the license. You may obtain a copy of the license at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the license is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."
          }
        ]
      }
    }
  ]
}
```

There is a wealth of information about the patient here, but the main field of interest for purposes of this exercise is the id. Specified under the “resource” element of a member of the “entry” collection, this is the unique identifier of this particular patient record within the PIT. It is what other records will use to refer to this patient and thus can be used to search for only records relevant to this patient.

Query 2 – Locating EOB Records for the Patient (Silvercare)

Once Terry’s ID is known, it is possible to locate his explanation of benefits records under the *ExplanationOfBenefits* resource. This is accomplished by first selecting the patient option in the search criteria dropdown, then entering the ID from the previous query in the Resource ID field. See below for an example.

Resources

ExplanationOfBenefit 262696

Coverage 9242

Patient 1435

Search
Queries
CRUD Operations
Tags

Search

Search Parameters
Optionally add parameter(s) to the search

+
patient - The reference to the patient
155

This should produce a complete set of Terry's EOB records for this PIT.

Result Body JSON bundle (403055 bytes)	Bundle contains 20 / 333 entries ◀ Prev Page Next Page ▶	
	ID	Updated
	Read Update ExplanationOfBenefit/1458/_history/1	2020-04-17 20:59:07
	Read Update ExplanationOfBenefit/1553/_history/1	2020-04-17 21:05:58
	Read Update ExplanationOfBenefit/1596/_history/1	2020-04-17 21:06:58
	Read Update ExplanationOfBenefit/2200/_history/1	2020-04-17 21:20:53
	Read Update ExplanationOfBenefit/2616/_history/1	2020-04-17 21:28:38
	Read Update ExplanationOfBenefit/2617/_history/1	2020-04-17 21:28:38
	Read Update ExplanationOfBenefit/2847/_history/1	2020-04-17 21:32:43
	Read Update ExplanationOfBenefit/3105/_history/1	2020-04-17 21:36:50
	Read Update ExplanationOfBenefit/3526/_history/1	2020-04-17 21:42:42
	Read Update ExplanationOfBenefit/4005/_history/1	2020-04-17 21:49:14
	Read Update ExplanationOfBenefit/4358/_history/1	2020-04-17 21:53:38
	Read Update ExplanationOfBenefit/4763/_history/1	2020-04-17 21:58:51

Raw Message

```
{
  "resourceType": "Bundle",
  "id": "e6327297-5bd7-4e93-a276-fcb8570b17f0",
  "meta": {
    "lastUpdated": "2020-05-15T15:24:32.459+00:00"
  },
  "type": "searchset",
  "total": 333,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/ExplanationOfBenefit?_pretty=true&patient=155"
    },
    {
      "relation": "next",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir?_getpages=e6327297-5bd7-4e93-a276-fcb8570b17f0&_getpages=20&_count=20&_pretty=true&_bundletype=searchset"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/ExplanationOfBenefit/1458",
      "resource": {
        "resourceType": "ExplanationOfBenefit",
        "id": "1458",
        "meta": {
          "extension": [
            {
              "url": "http://hl7.org/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#kU7KJCKB7xr6ZuT4"
            }
          ]
        },
        "versionId": "1",
        "lastUpdated": "2020-04-17T20:59:07.000+00:00"
      },
      "extension": [

```

Query 3 – Finding Coverage information for the Patient

Coverage information can be found in a very similar method by using the *beneficiary* option with the ID we used for Query 2.

The screenshot shows the FHIR Search interface. On the left, a sidebar lists resources: Resources (highlighted), ExplanationOfBenefit (262696), Coverage (9242), Patient (1435), and Account. The main area has tabs for Search, Queries, CRUD Operations, and Tags. The Search tab is active, showing a search bar with a magnifying glass icon and a 'Search' button. Below the search bar, the 'Search Parameters' section is visible, with a dropdown menu set to 'policy-holder - Reference to the policyholder' and a text input field containing '155'. The 'Includes' section below it states: 'Also include resources which are referenced by the search results'.

This should produce Terry's coverage information for this particular PIT.

Result Body JSON bundle (227213 bytes)	Bundle contains 11 / 11 entries	
	ID	Updated
	Read Update Coverage/156/_history/1	2020-04-17 20:52:25
	Read Update Coverage/4977/_history/1	2020-04-17 22:01:37
	Read Update Coverage/25783/_history/1	2020-04-18 02:32:55
	Read Update Coverage/49720/_history/1	2020-04-18 07:49:15
	Read Update Coverage/74189/_history/1	2020-04-18 13:12:50
	Read Update Coverage/100934/_history/1	2020-04-18 19:43:52

Raw Message

```
{
  "resourceType": "Bundle",
  "id": "96aba538-2d78-4b56-8add-ac44068bf689",
  "meta": {
    "lastUpdated": "2020-05-15T15:25:56.831+00:00"
  },
  "type": "searchset",
  "total": 11,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinterooperand.com/silvercare/fhir/Coverage?_pretty=true&policy-holder=155"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinterooperand.com/silvercare/fhir/Coverage/156",
      "resource": {
        "resourceType": "Coverage",
        "id": "156",
        "meta": {
          "extension": [
            {
              "url": "http://hl7.org/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#5TGdZNLlyUPqqpl"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:52:25.000+00:00"
        },
        "extension": [
          {
            "url": "http://mihin.org/extension/copyright",
            "valueString": "Copyright 2014-2020 Michigan Health Information Network Shared Services. Licensed under the License, Version 2.0 (the 'License'); you may not use this file except in compliance with the License. You may copy of the license at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the license is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."
          },
          {
            "url": "http://hl7.org/fhir/StructureDefinition/coverage-rxbin",
            "valueString": "612030"
          }
        ]
      }
    }
  ]
}
```

Scenario 2: Navigating Co-insurance

Terry has secondary insurance to complement his Medicare. How can we make it easier to see what each insurance is paying, and how much Terry is responsible for?

Action: Explore ways to showcase a patient's medical insurance information, review coverage, and allow them to review their EOBs for their secondary insurance.

Precondition: Be able to utilize the queries listed below in the Better Health Insurance PIT in IOL.

Success Criteria: Successfully locate and identify the patient's Medical Insurance and EOB's.

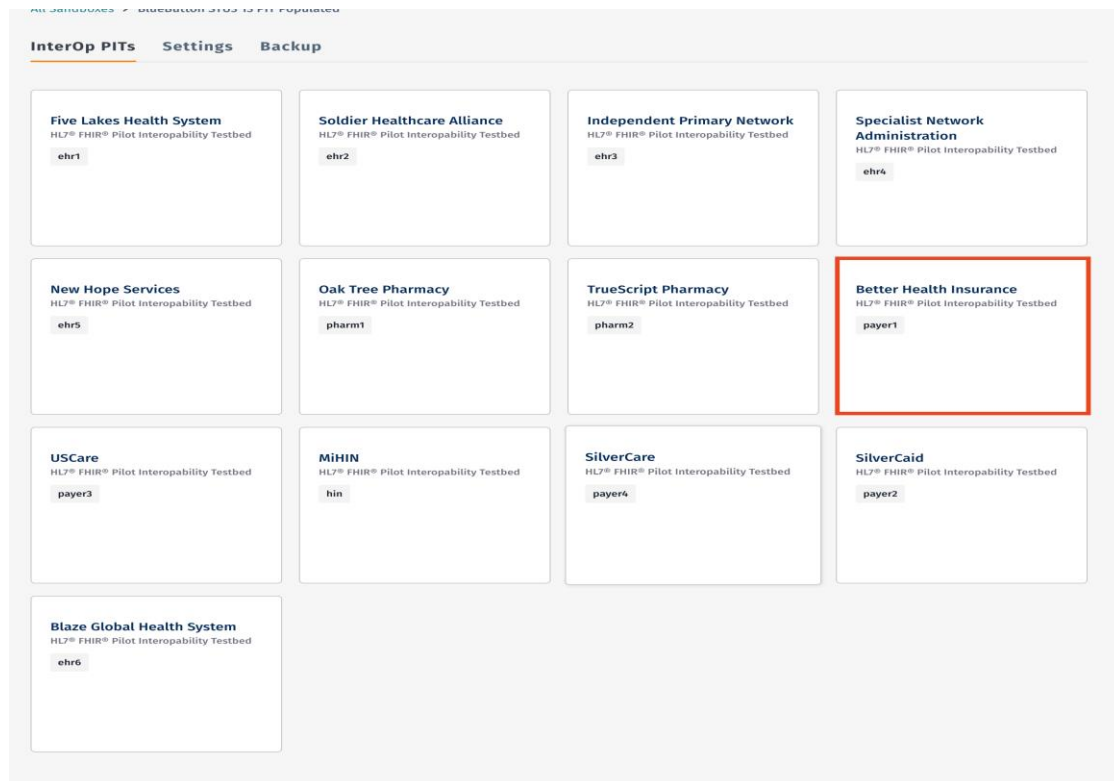
Examples- Possible Queries:

1. Find a patient ID using SSN
2. Locate the patient's EOB's using the *ExplanationOfBenefits* resource.
3. Review the patient's coverage information in the *Coverage* resource

The steps for this scenario are very similar to those for Scenario 1. First, we retrieve the patient's ID using his SSN, then we locate and explore his EOB and Coverage records. The primary difference for this scenario is that we will be performing the search on two payer PIT's, and using the resulting data to build a more complete picture of Terry's insurance profile.

Query 1 – Finding A Patient By SSN

As in Scenario 1, we begin by locating Terry's ID on the payer PIT. This time in addition to the *Silvercare* PIT, we will be searching *Better Health Insurance* for Terry's information. This can be accomplished by first navigating to the Better Health insurance PIT from the sandbox dashboard.



We use the same social security number (000003512) with in the *code* field of the *identifier* option, as pictured below.

The screenshot shows a search interface with a left sidebar listing resources: Claim (276470), ClaimResponse (276470), ExplanationOfBenefit (276470), Coverage (17703), Patient (3910), and QuestionnaireResponse (1076). The 'Patient' resource is highlighted. The main panel has tabs for Search, Queries, CRUD Operations, and Tags. A search bar is present. Below it, 'Search Parameters' are configured: 'identifier - A patient identifier' is selected from a dropdown, and the 'Code' field is set to '00003512'. There are also 'Matches', 'System', and '(opt)' fields.

We can then make a note of Terry's ID on this second payer PIT. As you can see below, this ID is different from the one in the *Silvercare* PIT, and indeed each PIT has a unique ID for each patient record.

➤ Request	GET https://dev-jtx37.devinteropland.com/better-health-insurance/fhir/Patient?identifier=000003512&_pretty=true				
Request Headers	Accept-Charset: utf-8 Authorization: Basic aW50ZXJvcF9waXQ6UUNxdWhyblkzcDFSSDRlZTlsejE3cmlyZ2VjeWo0TW9BTFFp0 Accept: application/fhir+xml;q=1.0, application/fhir+json;q=1.0, application/xml+fhir;q=0.9, application/json+fhir;q=0.9 User-Agent: HAPI-FHIR/4.2.0 (FHIR Client; FHIR 3.0.2/DSTU3; apache) Accept-Encoding: gzip				
⬅ Response	✓ HTTP 200				
Response Headers	x-request-id: juUbJQiy26IfQPMF date: Mon, 04 May 2020 14:52:12 GMT last-modified: Mon, 04 May 2020 14:52:12 GMT transfer-encoding: chunked x-powered-by: HAPI FHIR 4.2.0 REST Server (FHIR Server; FHIR 3.0.2/DSTU3) connection: keep-alive content-type: application/fhir+json; charset=UTF-8				
Result Body JSON bundle (4646 bytes)	<div> <div>Bundle contains 1 / 1 entries</div> <table> <thead> <tr> <th>ID</th><th>Updated</th></tr> </thead> <tbody> <tr> <td> <div>Read Update</div> Patient/224793/_history/1 </td><td>2020-04-18 13:12:50</td></tr> </tbody> </table> </div> <div>Raw Message</div> <pre> { "resourceType": "Bundle", "id": "46d77310-0364-4db6-a0f5-af1262a1c3b0", "meta": { "lastUpdated": "2020-05-04T14:52:12.023+00:00" }, "type": "searchset", "total": 1, "link": [{ "relation": "self", "url": "https://dev-jtx37.devinteropland.com/better-health-insurance/fhir/Patient?_pretty=true&identifier=000003512" }], "entry": [{ "fullUrl": "https://dev-jtx37.devinteropland.com/better-health-insurance/fhir/Patient/224793", </pre>	ID	Updated	<div>Read Update</div> Patient/224793/_history/1	2020-04-18 13:12:50
ID	Updated				
<div>Read Update</div> Patient/224793/_history/1	2020-04-18 13:12:50				

To navigate the Explanation of Benefits resource and review the patient's coverage information, simply repeat the steps for Queries 2 and 3 from Scenario 1 in the Better Health Insurance PIT with the appropriate patient ID for Better Health Insurance.

Scenario 3: Comparing Medication Prices

Action: Pull active meds for patients with Medicaid from Pharmacy PIT and coverage from payer PIT, then cross reference with local pharmacies to identify the best price.

Precondition: Be able to utilize the queries listed below and access third-party prescription services.

Success Criteria: Successfully compare prices of an active medication for one of the patients in the PIT.

Example: Possible Queries:

1. Finding the Patient's ID on an EHR PIT
2. Finding Medication Requests for a patient
3. Finding a Patient by SSN in the Silvercare PIT
4. Finding a Patient's Prescriptions in Their EOB Records

Query 1 – Finding the Patient's ID on an EHR PIT (Soldier Healthcare Alliance)

It is also possible to search for medication requests on an EHR PIT. First, like above, we must find the patient's ID on that PIT using their SSN and the *identifier* search option.

CarePlan 9287

MedicationStatement 6532

Patient 2422

Goal 1518

DocumentReference 590

Device 436

Search Queries CRUD Operations Tags

Search

Search Parameters Optionally add parameter(s) to the search

+ Identifier - A patient identifier

Matches System (opt) Code 000003433

Includes Also include resources which are referenced by the search results

☐ Patient:general-practitioner ☐ Patient:link ☐ Patient:organization

This search returns one result, and provides us with the patient's ID on this PIT.

Result Body
JSON bundle
(4650 bytes)

Bundle contains 1 / 1 entries

ID	Updated
<div><div>ReadUpdate</div>Patient/46/_history/1</div>	2020-04-17 20:58:55

Raw Message

```
{
  "resourceType": "Bundle",
  "id": "00fa3cee-3e8c-4bd0-8154-0722335c973d",
  "meta": {
    "lastUpdated": "2020-05-05T17:55:34.345+00:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/soldier-healthcare-alliance/fhir/Patient?_pretty=true&identifier=00003433"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/soldier-healthcare-alliance/fhir/Patient/46",
      "resource": {
        "resourceType": "Patient",
        "id": "46",
        "extension": [

```

This Identifier will be used in Query 2 to help locate medication requests.

Query 2 – Finding Medication Requests for the patient (Soldier Healthcare Alliance)

A medication request is initiated when a physician or provider orders a new medication. Finding medication requests for the patient can be done by navigating to the *MedicationRequest* resource and selecting the *patient* option. Use the ID acquired in Query 1 and click “Search.”

DiagnosticReport 119336

Condition 23920

Immunization 22022

MedicationRequest 16170

CarePlan 9287

SearchQueriesCRUD OperationsTags

Q Search

Search Parameters Optionally add parameter(s) to the search

+

patient - Returns prescriptions for a specific patient

46

Includes Also include resources which are referenced by the search results

This returns the medication information, which we can then be cross-reference with a third-party service such as GoodRx to provide useful price comparisons.

Result Body
 JSON bundle
 (12557 bytes)

Bundle contains 4 / 4 entries

ID	Updated
<div>Read Update</div> MedicationRequest/64/_history/1	2020-04-17 20:58:55
<div>Read Update</div> MedicationRequest/895/_history/1	2020-04-17 21:05:56
<div>Read Update</div> MedicationRequest/897/_history/1	2020-04-17 21:05:56
<div>Read Update</div> MedicationRequest/89952/_history/1	2020-04-18 01:45:31

Raw Message

```

{
  "resourceType": "Bundle",
  "id": "7892b265-d01c-489f-924d-110486526552",
  "meta": {
    "lastUpdated": "2020-05-05T18:00:57.349+00:00"
  },
  "type": "searchset",
  "total": 4,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/soldier-healthcare-alliance/fhir/MedicationRequest?_pretty=true&patient=46"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/soldier-healthcare-alliance/fhir/MedicationRequest/64",
      "resource": {
        "resourceType": "MedicationRequest",
        "id": "64",
        "meta": {
          "extension": [
            {
              "url": "http://hl7.org/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#700D0F4LcplWHCA"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:58:55.000+00:00",
          "profile": "http://hl7.org/fhir/us/hedis/StructureDefinition/hedis-medicationrequest"
        },
        "extension": [
          {
            "url": "http://mihin.org/extension/copyright",
            "valueString": "Copyright 2014-2020 Michigan Health Information Network Shared Services. Licensed under the Apache License, Version 2.0 (the 'License'); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."
          }
        ],
        "identifier": [
          {
            "use": "official",
            "system": "http://mihin.org/fhir/sid/elementId",
            "value": "e9e86896-2b0f-4585-8511-c56927350668"
          }
        ],
        "status": "active",
        "medicationReference": {
          "reference": "Medication/16",
          "display": "Insulin human, isophane 70 UNT/ML / Regular Insulin, Human 30 UNT/ML Injectable Suspension [Humulin]"
        },
        "subject": {
          "reference": "Patient/46",
          "display": "Claudia Rosario Perkins"
        },
        "encounter": {
          "reference": "Encounter/48",
          "display": "T1_Diabetes_Emergency_Encounter"
        },
        "authoredOn": "2016-01-15T00:00:00+00:00",
        "requester": {
          "reference": "Practitioner/47",
          "display": "Minnie Arroyo Owens MD"
        },
        "reasonCode": [
          {
            "coding": [
              {
                "system": "http://snomed.info/sct",
                "code": "402216091",
                "display": "Diabetes mellitus type 2"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

Medication Information

Patient information

Query 3 -Finding a Patient by SSN in the Silvercare and EHR PIT

Once again, we must locate the patient's ID on the payer PIT. This time in addition to the *Silvercare* PIT, we will also be searching an EHR PIT (Soldier Healthcare Alliance) for *MedicationRequest* records, which we can then use a third-party site to attempt to compare prices.

We use the patient's social security number (000003433) in the *code* field of the *identifier* option, as pictured below.

ExplanationOfBenefit 187100

Coverage 21424

Patient 5932

Account

ActivityDefinition

Search Queries CRUD Operations Tags

Search

Search Parameters Optionally add parameter(s) to the search

+ identifier - A patient identifier

Matches System (opt) Code 000003433

Includes Also include resources which are referenced by the search results

← Response ✓ HTTP 200

Response Headers

x-request-id: ak3ZyTYi5ITf0giB
date: Tue, 05 May 2020 17:29:40 GMT
last-modified: Tue, 05 May 2020 17:29:40 GMT
transfer-encoding: chunked
x-powered-by: HAPI FHIR 4.2.0 REST Server (FHIR Server; FHIR 3.0.2/DSTU3)
connection: keep-alive
content-type: application/fhir+json;charset=UTF-8

Result Body
JSON bundle
(4460 bytes)

Bundle contains 1 / 1 entries

ID	Updated
Read Update Patient/861/_history/1	2020-04-17 20:52:19

Raw Message

```
{
  "resourceType": "Bundle",
  "id": "56d72af8-5430-4ac1-9b15-b95a73083ab2",
  "meta": {
    "lastUpdated": "2020-05-05T17:29:40.353+00:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteroplant.com/silvercaid/fhir/Patient?_pretty=true&identifier=000003433"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteroplant.com/silvercaid/fhir/Patient/861",
      "resource": {
        "id": "861",
        "extension": [

```

We can then make a note of the ID for use in exploring EOB records for this patient

Query 4- Finding a Patient's Prescriptions in Their EOB Records

By using the ID gained in the previous query, it is possible to search the patient's EOB records for prescriptions. Select the *patient* option, and enter the patient ID as seen below.

Server Home/Actions

Resources

ExplanationOfBenefit 187100

Coverage 21424

Patient 5932

Search Queries CRUD Operations Tags

Search

Search Parameters Optionally add parameter(s) to the search

+ patient - The reference to the patient

861

Includes Also include resources which are referenced by the search results

Identifying prescription-based EOB's can be done by searching for “**rxnorm**” in the coding section of the record

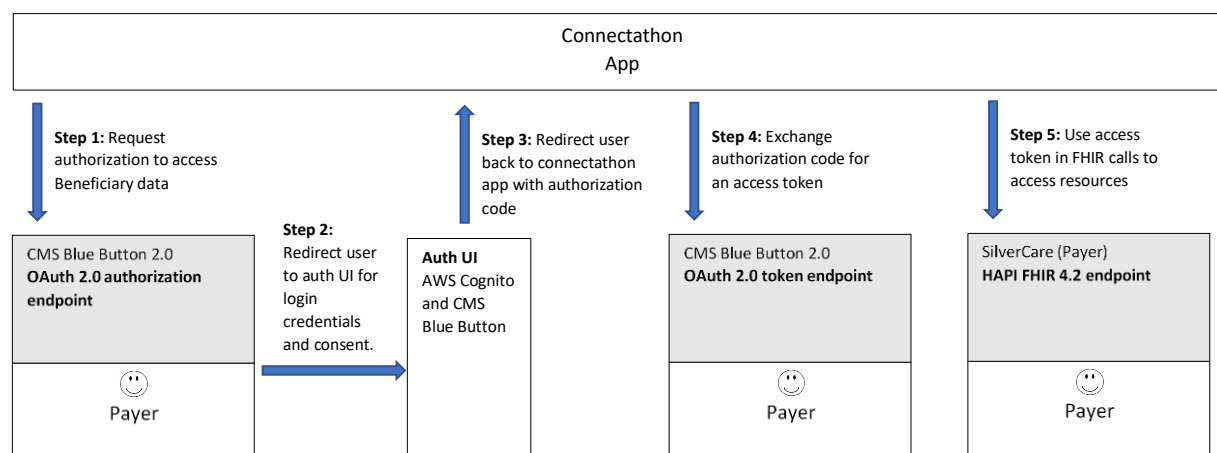
```
{
  "sequence": 2,
  "careTeamLinkId": [
    0
  ],
  "service": {
    "coding": [
      {
        "system": "http://www.nlm.nih.gov/research/umls/rxnorm",
        "code": "757894",
        "display": "Jolivet 28 Day Pack"
      }
    ]
  },
  "locationCodeableConcept": {
    "coding": [
      {
        "system": "https://bluebutton.cms.gov/resources/variables/line_place_of_srvcd",
        "code": "21",
        "display": "Inpatient Hospital"
      }
    ]
  }
}
```

This can be used to determine that this is a prescription EOB

The name of the prescription

Appendix A: Steps for Authorization

CMS Blue Button 2.0 Architecture Diagram
Access Beneficiary Data Scenario



Authorization: In order to query a patient's data as described in the included scenarios, we must complete the process of creating a Blue Button 2.0 application, and authorizing it to access of our sample Patient's data.


Note: For this setup, you will need the username and password of the patient that have been provided for your team.

Step 1 - Acquiring the IOL Sandbox Name and URL

First, log into IOL and navigate to the relevant sandbox. Select the Silvercare PIT from the dashboard, and you will be presented with an Overview tab similar to the one below. Note the name and URL of the sandbox for later use. The Basic Auth username and password can be used for bulk querying patient data if needed.

Overview
Capabilities Statement
SMART Apps
Visualizations

SilverCare


This is not a production server. Do not store any personal health or other confidential information here.

Link

https://dev-jtx37.devinteropland.com/silvercare/
Copy

Sandbox Name

Username

interop_pit
Copy

Basic Auth Username and Password

Password

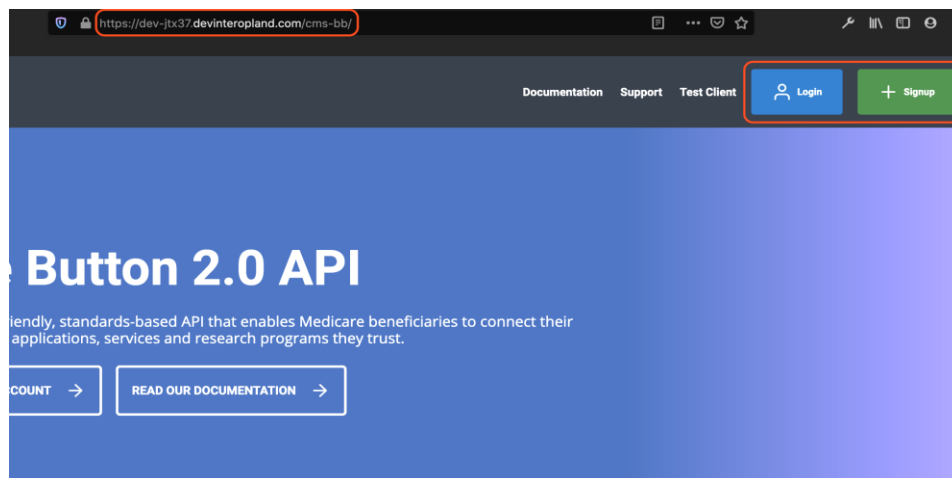
.....
Reveal
Copy

Basic Authentication Credentials

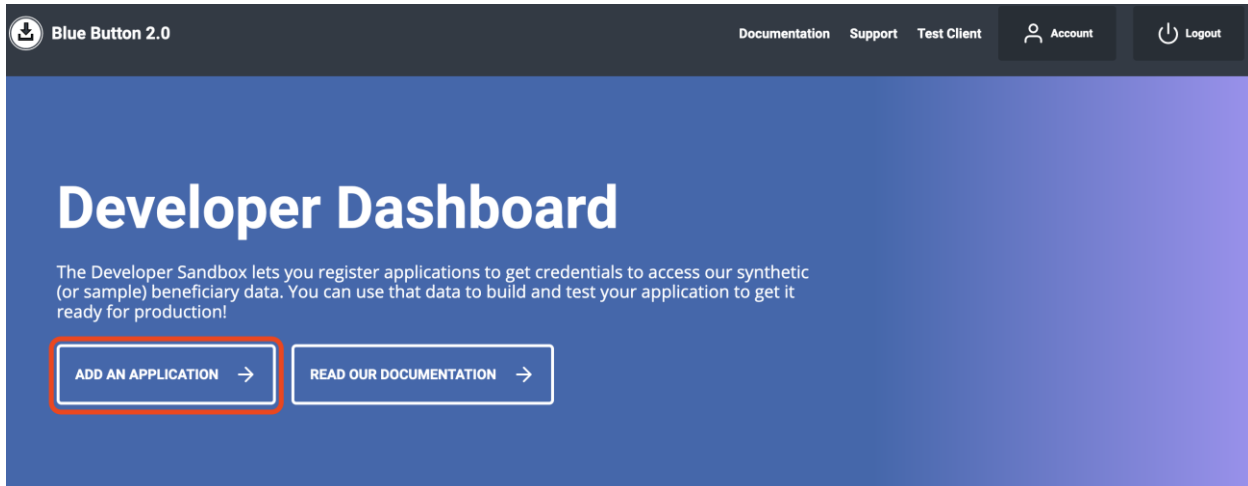
.....
Reveal
Copy

Step 2 - Creating a Blue Button Application

Now that we have the base URL for our sandbox, we can access the Blue Button 2.0 dashboard. Navigate to the address below (replace the sandbox name and domain with the values you acquired in step one). You should see the Blue Button 2.0 dashboard.



Next we will need to log in or create a new account. Select either option and fill in the form with your information. After you have submitted the form, you will arrive at the Developer Dashboard, where it's possible to create a new Blue Button application.



Click “Add an Application” and fill out the form as shown below. Choose any name for the application, and set the **Callback URLs / Redirect URIs** field to **<https://oauthdebugger.com/debug>**. This will allow us to inspect the responses to our authorization calls, and obtain a necessary code for use later steps.

App Details - Required Info

Application Name
IOL InterOpathon Blue Button Demo

OAuth - Client Type
Confidential

Authorization Grant Type*
Authorization code

Callback URLs / Redirect URIs
https://oauthdebugger.com/debug

Optional App Information

☒ Yes I have read and agree to the [API Terms of Service Agreement*](#)

Save Application

After creating the application, you will be directed to the application details page. The **App Credentials** section should feature a **Client ID** field, as well as a **Client Secret** field. Make a note of both values, as they will be crucial in later steps. For convenience, clicking each field will copy its value to your clipboard.

App Credentials

Client ID [Show/Hide Credentials](#)

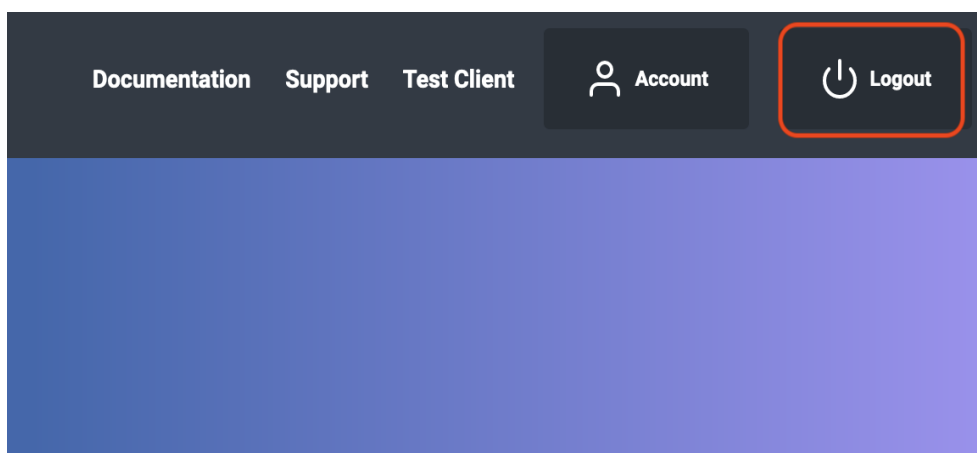
txUQN9YfxZgCCYkiHy4fdtzt2nmuvEt1lQAfmPih

Client Secret

dPlwfiual5DBUnybU7yp1YDa1hWJRKad4d2igTlxLmBKuP2M

Step 3 - Acquiring the Authorization Code

Now that we have a **Client ID** and **Client Secret** for our application, we need to log in as our sample patient and authorize the application to access the **Patient**, **ExplanationOfBenefits**, and **Coverage** records associated with that patient's account. First, log out of the Blue Button dashboard by pressing "Logout" in the top-right corner of the screen.



Next, navigate to oauthdebugger.com, where we will configure and run our authorization request. Fill out the form as shown below and press “Send Request”.

Authorize URI (required)

Redirect URI (required)

Client ID (required)

Scope (required)

State

Nonce

Response type (required)
☒ code ☐ token

Authorization code flow

The authorization server will respond with a **code**, which the client can exchange for tokens on a secure channel. This flow should be used when the application code runs on a secure server (common for MVC and server-rendered pages apps).

Response mode (required)
☐ query ☒ form_post ☐ fragment

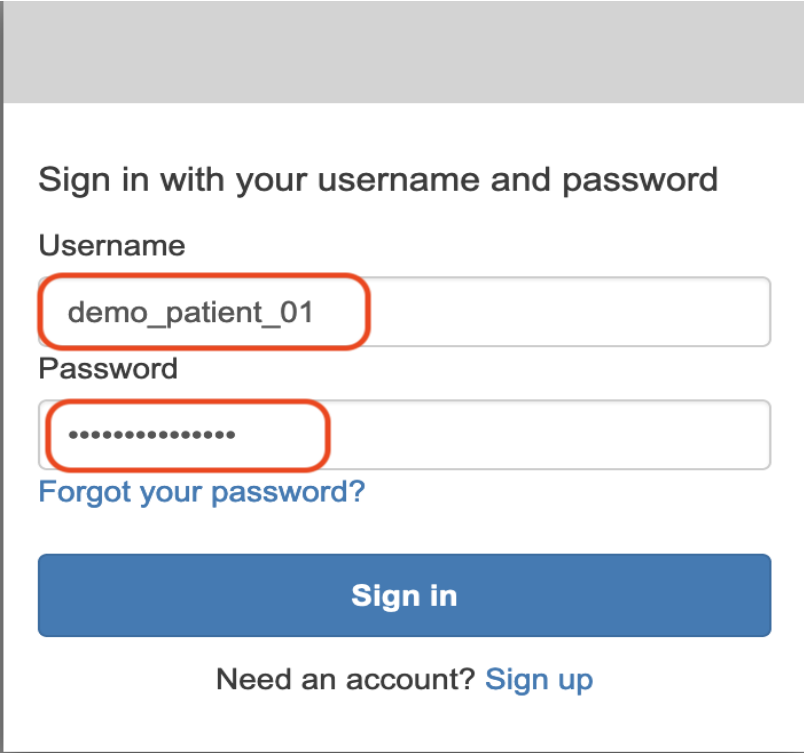
```
https://dev-jtx37.devinteroplاند.com/cms-bb/v1/o/authorize
?client_id=txUQN9YfxZgCCYkiHy4fdtzt2nmuvEt1lQAfmPih
&redirect_uri=https://oauthdebugger.com/debug
&scope=profile patient/Patient.read patient/ExplanationOfBenefit.read
patient/Coverage.read
&response_type=code
&response_mode=form_post
&nonce=1bt891gmqpd
```

SEND REQUEST

Note: For your convenience, the value of the **Scope** field is:

```
profile patient/Patient.read patient/ExplanationOfBenefit.read patient/Coverage.read
```

After sending the request, you will be prompted for a username and password. Since our goal is to authorize our new application to access our sample patient's records, we will use the provided username and password to log in as that patient. Enter the credentials and click "Sign in" to proceed.

A screenshot of a web-based sign-in form. At the top, a grey header bar is visible. Below it, the text "Sign in with your username and password" is centered. Underneath, the label "Username" is followed by a text input field containing "demo_patient_01". Below that, the label "Password" is followed by a text input field filled with dots. A blue link "Forgot your password?" is positioned below the password field. At the bottom of the form is a large blue button with the text "Sign in" in white. Below the button, the text "Need an account? Sign up" is displayed, with "Sign up" as a blue link. The entire form is enclosed in a thin black border.

Sign in with your username and password

Username

demo_patient_01

Password

.....

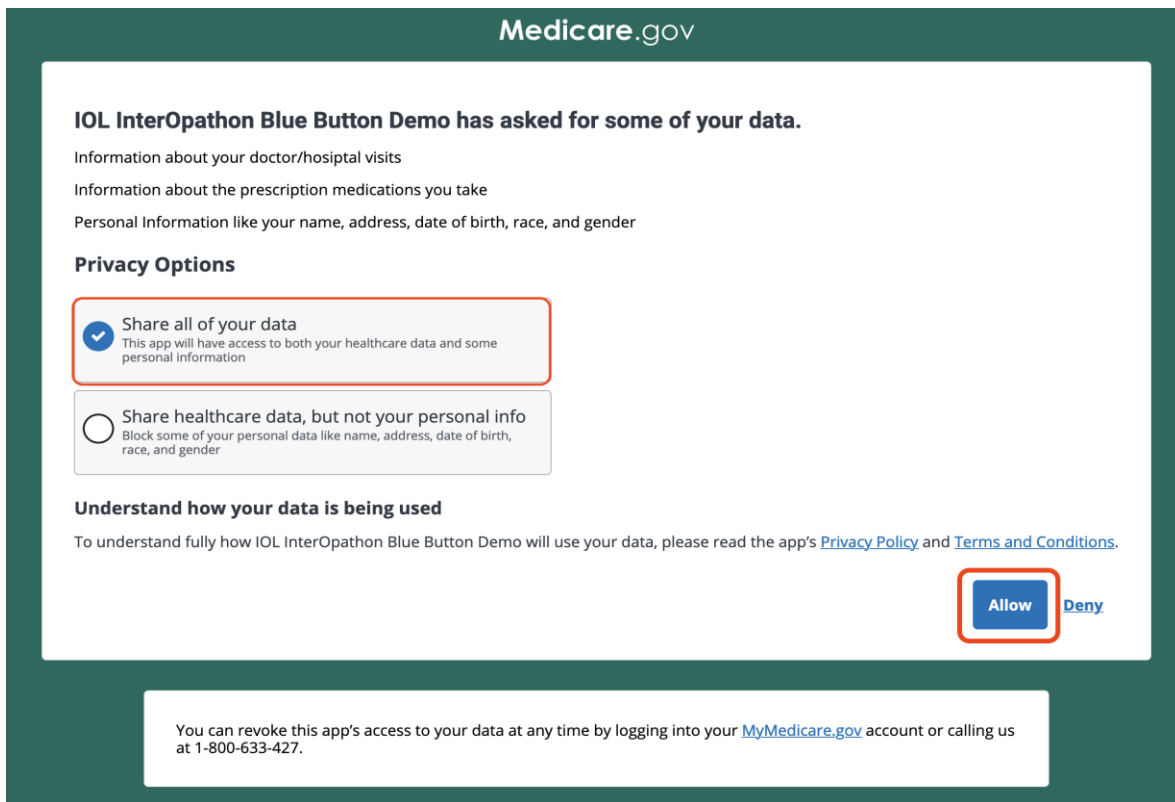
[Forgot your password?](#)

Sign in

Need an account? [Sign up](#)

After submitting the patient's credentials, you will be prompted to grant data access permissions. Select "Share all of your data" and click "Allow".

After granting access, you will be shown a response page, which should feature an **Authorization code** as seen in the image below.



The image shows a Medicare.gov authorization page. At the top, it says "Medicare.gov". Below that, it states "IOL InterOpathon Blue Button Demo has asked for some of your data." and lists the types of data requested: "Information about your doctor/hospital visits", "Information about the prescription medications you take", and "Personal Information like your name, address, date of birth, race, and gender". Under the heading "Privacy Options", there are two radio button options. The first option, "Share all of your data", is selected and highlighted with a red box; its description is "This app will have access to both your healthcare data and some personal information". The second option is "Share healthcare data, but not your personal info", with a description "Block some of your personal data like name, address, date of birth, race, and gender". Below this, under the heading "Understand how your data is being used", it says "To understand fully how IOL InterOpathon Blue Button Demo will use your data, please read the app's [Privacy Policy](#) and [Terms and Conditions](#)." At the bottom right, there are two buttons: "Allow" (highlighted with a red box) and "Deny". At the very bottom, a white box contains the text: "You can revoke this app's access to your data at any time by logging into your [MyMedicare.gov](#) account or calling us at 1-800-633-427."

We're also provided with a hint as to our next step. We will need to exchange this code for a bearer token for use in our patient record queries. Make note of the fields listed in the **Step 2** block. They will need to be correctly populated in order to acquire the token.

OAuth 2.0 <debugger/>

Test OAuth 2.0 requests and debug responses.

✓ Success!

↺ Start over

The flow was successful. The authorization server responded with an authorization code because the flow was started with the **code** response type.

Authorization code

CxhHhLWKB7nemD1hdnbQjr7wVHoXnu

Step 2: Exchange code for tokens

Now you need to exchange the authorization code for tokens using the token endpoint. We can't do this step for you because it involves your client secret.

```
POST {tokenEndpoint}
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=CxhHhLWKB7nemD1hdnbQjr7wVHoXnu&
client_id={clientId}&
client_secret={clientSecret}&
redirect_uri={redirectUri}
```

Step 4 - Exchanging the Authorization Code for a Bearer Token

By this point we have collected everything necessary for the token request. Open your API client of choice (we use Postman in this example) and prepare the request as demonstrated below. Substitute the address, code, client_id and client_secret with the values obtained in the previous step.

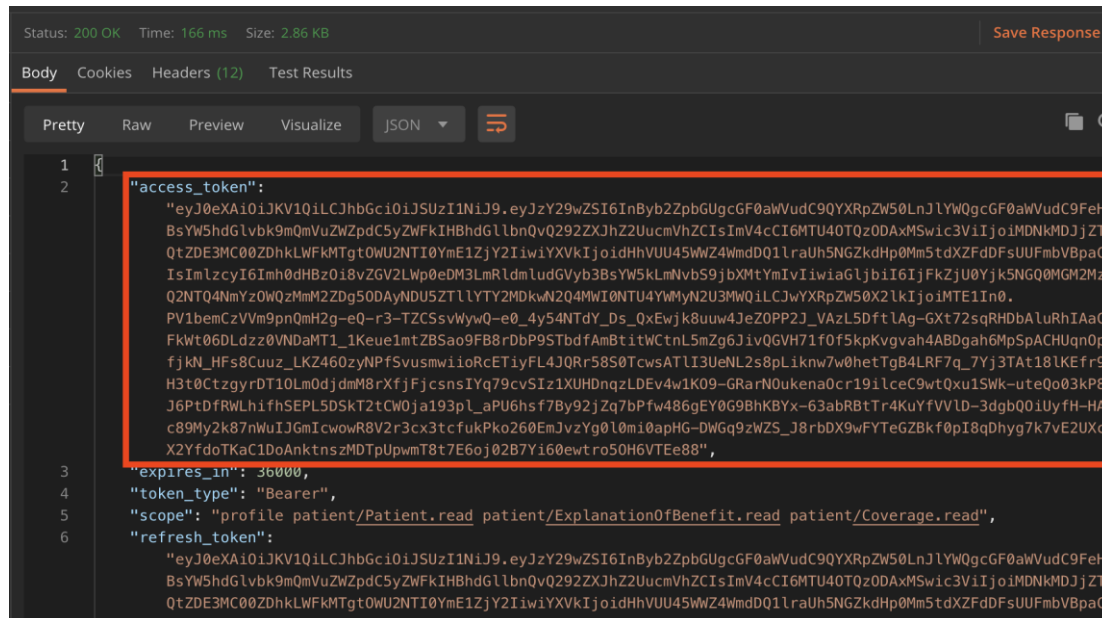
POST <https://dev-jtx37.devinteroplant.com/cms-bb/v1/o/token/>

Params Auth Headers (7) **Body** Pre-req. Tests Settings Cookies Code

x-www-form-urlencoded

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Content-Type ...	application/x-www-form-urlencoded ...			
<input checked="" type="checkbox"/>	grant_type	authorization_code ...			
<input checked="" type="checkbox"/>	code	CxhHhLWKB7nemD1hdnbQjr7wVHoXnu			
<input checked="" type="checkbox"/>	client_id	txUQN9YfxZgCCYkiHy4fdtzt2nmuvEt1lQAfmPih			
<input checked="" type="checkbox"/>	client_secret ...	dPlwfuaI5DBUnybU7yp1YDa1hWJRKad4d2igTlxLmBKuP2MdgSRTJj6ML...			
<input checked="" type="checkbox"/>	redirect_uri	https://oauthdebugger.com/debug ...			
	Key	Value	Description		

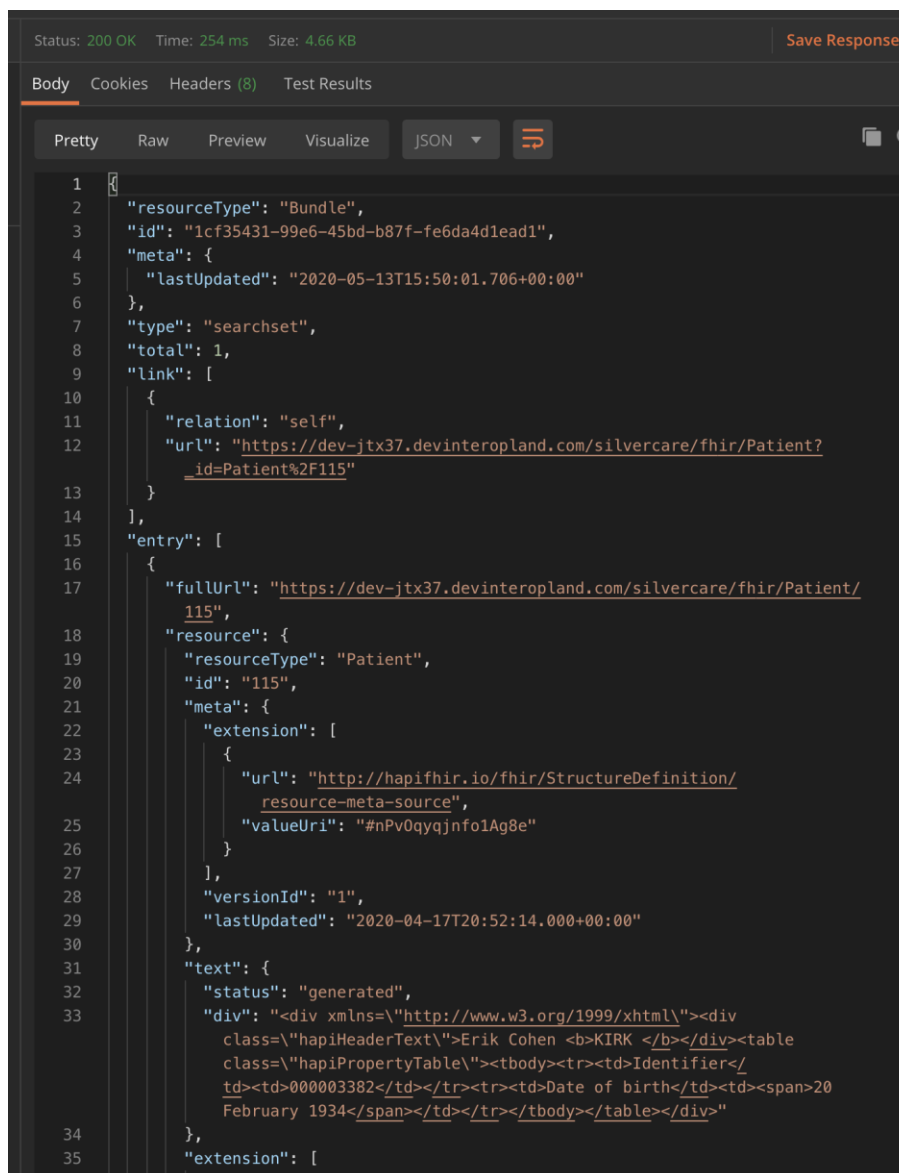
Send the request. The response body should be similar to the following:



```
1 {
2   "access_token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
4     BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
5     QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
6     IsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
7     Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
8     PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
9     Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
10    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
11    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
12    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
13    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
14    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
15  "expires_in": 3600,
16  "token_type": "Bearer",
17  "scope": "profile patient/Patient.read patient/ExplanationOfBenefit.read patient/Coverage.read",
18  "refresh_token":
19    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
20    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
21    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
22    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
23    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
24    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
25    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
26    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
27    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
28    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
29    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
30    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
31  "refresh_token":
32    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
33    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
34    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
35    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
36    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
37    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
38    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
39    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
40    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
41    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
42    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
43    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
44  "refresh_token":
45    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
46    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
47    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
48    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
49    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
50    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
51    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
52    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
53    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
54    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
55    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
56    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
57  "refresh_token":
58    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
59    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
60    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
61    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
62    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
63    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
64    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
65    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
66    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
67    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
68    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
69    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
70  "refresh_token":
71    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
72    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
73    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
74    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
75    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
76    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
77    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
78    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
79    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
80    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
81    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
82    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
83  "refresh_token":
84    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
85    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
86    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
87    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
88    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
89    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
90    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
91    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
92    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
93    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
94    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
95    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
96  "refresh_token":
97    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
98    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
99    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
100   aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
101   Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
102   PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
103   Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
104   fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
105   H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
106   J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
107   c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
108   X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
109  "refresh_token":
110    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
111    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
112    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
113    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
114    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
115    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
116    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
117    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
118    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
119    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
120    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
121    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
122  "refresh_token":
123    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
124    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
125    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
126    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
127    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
128    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
129    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
130    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
131    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
132    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
133    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
134    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
135  "refresh_token":
136    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
137    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
138    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
139    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
140    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
141    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
142    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
143    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
144    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
145    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
146    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
147    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
148  "refresh_token":
149    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
150    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
151    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
152    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
153    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
154    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
155    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
156    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
157    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8
158    J6PtDfRWLhifhSEPL5DSktZtCW0ja193pl_aPU6hsf7By92jZq7bPfw486gEY0G9BhKBYx-63abRBtTr4KuYfVVLID-3dgb00iUyfh-HA
159    c89My2k87nWuIJGmIcwowR8V2r3cx3tcfukPko260EmJvZy0l0mi0apHG-DWGq9zWZS_J8rbDX9wFyTeGZBkf0pI8qDhyg7k7vE2UXc
160    X2YfdoTKaC1DoAnktnszMDTpUpwmT8t7E6oj02B7Yi60ewtro50H6VTEe88",
161  "refresh_token":
162    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY29wZSI6InByb2ZpbGUgcGF0aWVudC90YXRpZW50LnJlYWQgcGF0aWVudC9FeH
163    BsYw5hdGlvbk9mQmVuZwZpdC5yZWFKIHBhdGllbnQvQ292ZXJhZ2UucmVhZCI6ImV4cCI6MTU4OTQzODAxMSwic3ViOiMDNkMDJjZT
164    QtZDE3MC00ZDhkLWFKMTgtOWU2NTI0YmE1ZjY2IiwiaXVkiOiJoidHhVUU45WmZ4WmdDQ1lraUhsNGZkdHh0Mm5tdXZfDFsUUFmbVBp
165    aCIsImIzcyI6Imh0dHBzOi8vZGV2LWp0eDM3LmRldmudGVyb3B5Yw5kLmNvbS9jbXN0YmIiIiwiaGljaCI6IjFkZjU0YjY5NGQ0MGZMz
166    Q2NTQ4NmYzOWQzMmM2ZDg5ODAyNDU5ZTllYTZMDkxwN2Q4MmI0NTU4YmMyN2U3MmQ1LCJwYXRpZW50X2lkIjoimTE1In0.
167    PV1bemCzVWm9pnQmH2g-eQ-r3-TZCSsvWywQ-e0_4y54NTdY_Ds_QxEwjK8uuw4Jez0PP2J_VAzL5DfttAg-GXt72sqRHDbaLuRhIAaC
168    Fkwt06DLdzz0VNDaMT1_1Keue1mtZBSao9FB8rDbP9STbdfAmBtitwCtnL5mZg6JivQGVH71f0f5kpKvgvah4ABDgah6Mp5pACHUqn0p
169    fjkN_HFs8Cuuz_LKZ460zyNPfSvusmwiioRcETiyFL4JQRr58S0TcwsATLI3UeNL2s8pLknw7w0hetTgB4LRF7q_7Yj3TAt18LKEfr9
170    H3t0CtzyrDT10Lm0djdM8rXfjFjcsnsIYq79cvSIz1XUHDnqzLDEv4w1K09-GRarN0ukena0cr19ilceC9wtQxu1SWk-uteQo03kP8

```

If successful, we should receive the appropriate data block for our sample patient, as shown in the following image.



```
1 {
2   "resourceType": "Bundle",
3   "id": "1cf35431-99e6-45bd-b87f-fe6da4d1ead1",
4   "meta": {
5     "lastUpdated": "2020-05-13T15:50:01.706+00:00"
6   },
7   "type": "searchset",
8   "total": 1,
9   "link": [
10    {
11      "relation": "self",
12      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient?_id=Patient%2F115"
13    }
14  ],
15  "entry": [
16    {
17      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient/115",
18      "resource": {
19        "resourceType": "Patient",
20        "id": "115",
21        "meta": {
22          "extension": [
23            {
24              "url": "http://hapi.fhir.io/fhir/StructureDefinition/resource-meta-source",
25              "valueUri": "#nPv0qyqjnfo1Ag8e"
26            }
27          ],
28          "versionId": "1",
29          "lastUpdated": "2020-04-17T20:52:14.000+00:00"
30        },
31        "text": {
32          "status": "generated",
33          "div": "<div xmlns='http://www.w3.org/1999/xhtml'><div class='hapiHeaderText'>Erik Cohen <b>KIRK </b></div><table class='hapiPropertyTable'><tbody><tr><td>Identifier</td><td>00003382</td></tr><tr><td>Date of birth</td><td><span>20 February 1934</span></td></tr></tbody></table></div>"
34        },
35        "extension": [
36
```

Note: Replace **Patient** with **ExplanationOfBenefit** or **Coverage** in the URL to facilitate completion of the Blue Button 2.0 scenarios.

Appendix B: Bulk Querying Patient Data

In order to query records for multiple users, you can use the Basic Auth username and password obtained in step 1.

The screenshot shows a REST client interface with a GET request to `https://{{Sandbox Name}}.{{Domain}}/silvercare/fhir/Patient`. The 'Auth' tab is selected and highlighted with an orange box. Below it, the 'TYPE' dropdown is set to 'Basic Auth' and also highlighted with an orange box. A descriptive text states: 'The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)'. In the authentication fields, the 'Username' is 'interop_pit' (highlighted with an orange box) and the 'Password' is masked with dots (highlighted with an orange box). A 'Show Password' checkbox is present and unchecked.

As you can see, our results include all Medicare patients. We can use the same authentication for ExplanationOfBenefits and Coverage records, and we will similarly see unfiltered records for those resources.

Appendix C - Sample Users

ID	SSN	Name	Username	Password
1	000003009	Georgia Burgess	gburgess	!lolconnect01
15	000003056	Tamara Murillo	tmurillo	!lolconnect02
25	000003087	Dwight Frey	dfrey	!lolconnect03
27	000003097	Jennie Moon	jmoon	!lolconnect04
29	000003111	Nicholas Butler	nbutler	!lolconnect05
31	000003112	Alicia Atkins	aatkins	!lolconnect06
33	000003123	Franklin Marks	fmarks	!lolconnect07
51	000003188	Perry Cortez	pcortez	!lolconnect08
53	000003189	Jeffrey Gregory	jgregory	!lolconnect09
55	000003193	Fernando Grant	fgrant	!lolconnect10
71	000003227	Leah Carson	lcarson	!lolconnect11
73	000003262	Keith Mooney	kmooney	!lolconnect12
75	000003265	Benjamin Dixon	bdixon	!lolconnect13
79	000003303	Mabel Avery	mavery	!lolconnect14
99	000003357	Dustin Neal	dneal	!lolconnect15

Judging Criteria

IGNITE

Interoperability: APIs and FHIR® Heat Up



INTEROPERABILITY
INSTITUTE



Alignment with Track	Helps to improve Inter-operability	Innovation & Creativity	Use of APIs	User Experience	Technical Difficulty	Presentation or Demo
25%	25%	15%	10%	10%	10%	5%
How aligned was the solution with one of the event Tracks?	Does the team clearly show how their solution could be used to improve interoperability?	Did the team create something that has not already been created? Is it unique?	Did the team use APIs available to create a solution?	What is the wow factor? Would others be impressed by what was built? How easy is the solution to use?	Is the project technically impressive / complex? Is it remarkable that a team created this solution in the time allowed?	Was the presentation or demo well put together? Did the team seem prepared? How well did they explain the problem and solution? (only judge on content, not video quality)

INTEROPATHON | 2020